# CS404FZ

## Assignment 1

**Submission Deadline: 11:59 pm (Beijing Time), 21 Dec, 2025**

## Part A: CNN for Image Recognition (40 marks)

In this assignment, you will design, implement, train, and evaluate a simple Convolutional Neural Network (CNN) for image classification. You are free to choose the architecture of your model and explore different training parameters.

You must train and evaluate your model on both the MNIST and CIFAR-10 datasets and compare their performance.

### 1. CNN Model Design (10 points)

Develop a simple CNN for image recognition. You may freely choose the network architecture, including the number of layers, filters, kernel sizes, and activation functions. Your model must handle both grayscale (MNIST) and colour (CIFAR-10)

### 2. Training and Parameter Exploration (10 points)

Train your model on both datasets using training parameters of your choice. Record the training time and investigate the effect of at least two training parameters (e.g., learning rate, batch size, number of epochs).

### 3. Evaluation and Dataset Comparison (10 points)

Evaluate your model using appropriate metrics such as accuracy and per-class performance (e.g., F1-score). Compare and discuss the performance differences between MNIST and CIFAR-10.

### 4. Architecture Comparison (5 points)

Implement and test a second CNN architecture and compare it with your first model in terms of training time and performance.

### 5. Visualisation (5 points)

Visualise your results using examples of correct and incorrect predictions, and optionally include confusion matrices or feature maps.

### Submission Requirement

You must submit a Python Jupyter Notebook containing the model implementation, training process, evaluation results, and visualisations.

## Part B: People Counting Using Pre-trained CNN Models (40 marks)

In this task, you will develop a people-counting system that detects and counts the number of people in a given image. You must use at least two classic pre-trained CNN-based object detection models.

**1. Model Selection and Setup (10 points)**

Select and implement at least two classic CNN-based detection models (e.g., YOLO, SSD, Faster R-CNN).

You should:

- Load the pre-trained models correctly
- Explain why you selected these models

**2. People Detection and Counting (10 points)**

Develop a Python program that:

- Takes an input image
- Detects all people in the image
- Outputs the total number of people detected

**3. Dataset Selection and Testing (10 points)**

Test your models using at least one suitable public human detection dataset.

Suggested datasets (you may choose one):

- COCO Dataset (recommended – includes a "person" class)
- PASCAL VOC
- CrowdHuman Dataset

You should:

- Describe the dataset used
- Explain why it is suitable for this task

**4. Model Comparison and Performance Discussion (5 points)**

Compare the two models in terms of:

- Detection accuracy
- False positives / missed detections

Provide a brief discussion of their strengths and limitations.

**5. Visualisation of Results (5 points)**

Visualise the detection results by:

- Drawing bounding boxes around detected people
- Displaying the final people count on the image

**Submission Requirement:**

You must submit a Python Jupyter Notebook that includes the model implementation, people detection and counting results, a description of the dataset used, a comparison of the selected models, and appropriate visualisations of the detection outcomes.

## Part C: Final Submission (10 marks)

You must submit a single ZIP file containing the Jupyter Notebook from Part A, the Jupyter Notebook from Part B, and a README.md file. The ZIP file must be named in the following format:

**{your_name-your_MU_ID}.zip**

The README file should clearly explain how to run the code, including environment setup, required libraries, and dataset preparation steps.